Center for Advanced Computation

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
URBANA, ILLINOIS 61801

214-14

CAC Document No. 71

THE QUADRATIC ASSIGNMENT PROBLEM

By

George B. Purdy

February 1973

CAC Document No. 71

THE QUADRATIC ASSIGNMENT PROBLEM

By

George B. Purdy

Center for Advanced Computation
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

February 1973

This work was supported in part by the Advanced Research Projects Agency of the Department of Defense and was monitored by the U.S. Army Research Office-Durham under Contract No. DAHC04-72-C-0001.

ABSTRACT

In this report we discuss some seemingly reasonable approaches to the quadratic assignment problem, and we give some evidence from automata theory that the problem is insoluble.

INTRODUCTION

We present here a discussion of some reasonable approaches to the quadratic assignment problem and an incomplete proof of its insolubility. For the former we reduce the problem to an integer programming problem, for the latter we relate the problem to some very difficult graph problems that are believed to be non-computable in any reasonable sense.

We shall define the quadratic assignment problem to be the following: Let A, B be given symmetric matrices with non-negative elements. Determine the permutation matrix P such that $tr(APBP^T)$ is maximized, where tr(A) denotes the trace of A.

Let π be a permutation on $\{1, \ldots, n\}$, and let $P = \{p_{ij}\}$ be the permutation matrix $p_{ij} = \delta_{\pi(i)j}$. Then

$$tr(APB^TP^T) = \sum_{r,s} a_{rs}\, b_{\pi(r)\pi(s)} \qquad (1).$$

Identity (1) is easily verified:

$$(APB^TP^T)_{ij} = \sum_{r,s,t} a_{ir}\, p_{rs}\, b_{ts}\, p_{jt}$$

But $p_{rs} = \delta_{\pi(r)s}$ is unity if $s = \pi(r)$ and zero otherwise; hence

$$(APB^TP^T)_{ij} = \sum_{r} a_{ir}\, b_{\pi(j)\pi(r)}$$

and (1) follows.

Thus we see that the quadratic assignment problem may also be formulated as follows: Given symmetric matrices A and B, with non-negative elements, find the permutation π which maximizes

$$\sum_{r,s} a_{rs}\, b_{\pi(r)\pi(s)}.$$

Since the number of possible permutations is n! for n by n matrices and since $20! \cong 2.4 \times 10^{18}$ and $30! \cong 2.6 \times 10^{32}$ it is not practical to solve the quadratic assignment problem by enumerating all possible sums unless n is very small.

The quadratic assignment problem can be stated as a boolean (i.e., zero-one) integer programming problem - a fact which follows from theorems one and two. Theorem one is of interest in itself. It states that the quadratic assignment problem on matrices A and B can be expressed as a boolean quadratic programming problem with objective function $\underline{x}^T Q \underline{x}$, where Q is $B \otimes A$, the Kronecker product of A and B (also called tensor product).

Theorem 1

Let A and B be n x n real matrices. Let $Q = B \otimes A$, the Kronecker product of B and A.

Then the maximum of

$$\operatorname{tr}\,(PAP^T B^T)$$

taken over all permutation matrices P is equal to the maximum of

$$\underline{x}^T\, Q\, \underline{x}$$

taken over all boolean (zero-one) vectors $\underline{x} = (x_1, x_2, \ldots, x_{n^2})$ subject to the 2n linear constraints

$$\sum_{i=1}^{n} x_{n(i-1)+j} = 1 \qquad 1 \le j \le n$$

and

$$\sum_{i=1}^{n} x_{n(i-1)+j} = 1 \qquad 1 \le i \le n.$$

Proof   Let A, B be n x n matrices. Then

$$\mathrm{tr}\,(PAP^TB^T) = \sum_i (PAP^TB^T)_{ii}$$

$$= \sum_i \sum_{rst} p_{ir}\, a_{rs}\, p^T_{st}\, b^T_{ti}$$

$$= \sum_{irst} p_{ir}\, a_{rs}\, p_{ts}\, b_{it}\,; \quad \text{let}$$

$\underline{x} = (x_1, \ldots, x_{n^2}),\ x_{n(i-1)+r} = p_{ir}$ for

$1 \le i,r \le n.$

Let us write $q_{ij} = q\,[i,j]$, where $Q = B \otimes A$. Then

$q\,[n(i-1) + r,\ n(j-1) + s] = b_{ij}\, a_{rs}, \quad 1 \le i,j,r,s \le n$, and

$$\sum_{irst} p_{ir}\, a_{rs}\, p_{ts}\, b_{it}$$

$$= \sum_{irst} x_{n(i-1)+r}\ q[n(i-1)+r,\ n(t-1)+s]\ x_{n(t-1)+s}$$

$$= \sum_{ij} x_i\ q[i,j]\ x_j = \sum_{ij} x_i\, q_{ij}\, x_j = \underline{x}^T Q\, \underline{x}\,.$$

The zero-one matrix P is a permutation matrix if and only if

$$\sum_i p_{ij} = 1 \qquad i \le j \le n$$

and

$$\sum_j p_{ij} = 1 \qquad 1 \le i \le n\,.$$

The resulting constraints on $\underline{x}$ are the ones claimed in the statement of the theorem.

Theorem 2

Let Q be an $n \times n$ real matrix, let A be an $m \times n$ matrix, and let $\underline{b}$ be an m-rowed vector, where $m \geq n$.

Then the maximum of $\underline{x}^T Q \underline{x}$ taken over all boolean (i.e. zero-one) vectors $\underline{x}$ subject to the constraint

$$A\underline{x} = \underline{b}$$

is equal to the maximum of

$$\underline{c} \cdot \underline{w}$$

taken over all boolean $\underline{w}$ subject to the constraint

$$Dy = \underline{d},$$

where $\underline{c}$ is a $(3n^2 - 2n)$-rowed vector, D is a $(2n^2 - 2n + m) \times (3n^2 - 2n)$ matrix, and $\underline{d}$ is a $(2n^2 - 2n + m)$-rowed vector.

Proof

In the function

$$\underline{x}^T Q \underline{x} = \sum x_i \, q_{ij} \, x_j$$

we wish to make the change of variable

$$(2) \qquad y_{ij} = x_i \, x_j \qquad 1 \leq i,j \leq n.$$

We need to introduce conditions on the y_{ij} to force them to be of the form (2).

Since all of our variables are zero-one, y_{ij} is of the form (2) if and only if

$$y_{ij} = y_{ii} \, y_{jj} \qquad (1 \leq i,j \leq n).$$

This is equivalent to

$$1 - y_{ij} = (1 - y_{ii}) \oplus (1 - y_{jj}) \qquad (1 \leq i,j \leq n),$$

where $\oplus$ denotes boolean addition.

Now boolean addition can be defined by adding variables and constraints. The only zero-one value z lying between $x + y$ and $\frac{1}{2}(x + y)$ is $x \oplus y$. Hence the equations

$$2z = x + y + u$$

$$z = x + y - v,$$

where u and v are zero-one variables, have the unique solution

$$z = x \oplus y .$$

We may now restate our original optimization problem as follows:

Maximize $\sum q_{ij} y_{ij}$ over all zero-one variables y_{ij}, u_{ij}, v_{ij} satisfying the constraints,

$$\sum_{j=1}^{n} a_{ij} y_{jj} = b_i \qquad (1 \le i \le m)$$

and

(3a) $$2(1 - y_{ij}) = (1 - y_{ii}) + (1 - y_{jj}) + u_{ij}$$

(3b) $$1 - y_{ij} = (1 - y_{ii}) + (1 - y_{jj}) - v_{ij}$$

$$1 \le i,j \le n \qquad i \ne j .$$

We observe that u_{ij} and v_{ij} are not needed when $i = j$. The total number of equations (constraints) in the new problem is $2n^2 - 2n + m$, and they are independent if the original m equations were. The number of variables is

$$3n^2 - 2n.$$

The problem is therefore of the type claimed in the statement of the theorem.

Now that theorems one and two have been proved, we are ready to make some remarks about this approach to the quadratic assignment problem. If we relax the boolean requirement in the boolean quadratic programming problem of Theorem 1, then we get an ordinary QP (quadratic programming) problem, and it is well known [1] that good algorithms exist when Q is negative definite or negative indefinite; otherwise, it is doubtful whether there exist good algorithms. Indeed, it was shown in [2] that Hilbert's tenth problem is expressible as a quadratic integer programming problem. (This differs from the problem of Theorem 1 in that the boolean variables are replaced by integer variables). Since Hilbert's tenth problem is known to be undecidable there can, of course, not be an algorithm. So, let us suppose that Q is negative definite. There is a theorem about Kronecker products which says that if λ_i $(1 \leq i \leq n)$ are the eigenvalues of A and μ_i $(1 \leq i \leq n)$ are the eigenvalues of B, then the eigenvalues of $B \otimes A$ are $\lambda_i \mu_j$ $(1 \leq i, j \leq n)$. Therefore, if A is negative definite so that all the λ_i are negative, and B is positive definite, so that all the μ_j are positive, then all the $\lambda_i \mu_j$ will be negative, and $B \otimes A$ will be negative definite, and the quadratic programming problem

$$
\left.
\begin{array}{lll}
\max \ \underline{x}^T Q \underline{x} & & \\
\text{subject to} \quad \sum_{i=1}^{n} x_{n(i-1)+j} = 1 & & 1 \leq j \leq n \\
\text{and} \quad \sum_{i=1}^{n} x_{n(i-1)+j} = 1 & & 1 \leq i \leq n
\end{array}
\right\} \quad (4)
$$

can be solved for real numbers x_i. Now we are interested in boolean solutions (4), and there is a theorem which says that the set of matrices P, $P_{ij} = x_{n(i-1)+j}$ satisfying (4) (called doubly stochastic matrices) is a convex set whose extreme points are precisely the n! permutation matrices of order n.

Thus a solution $\underline{x}$ to the QP problem will be boolean only if $\underline{x}$ is at a vertex of the polytope in n^2-dimensional space defined by (4). It is more typical, however, for such solutions to be interior points. The QP algorithms usually find a local maximum; the negative definiteness of Q implies that the local maximum is the global maximum.

If, on the other hand, we arrange to make Q positive definite - e.g., by choosing A and B to be positive definite, then any real solution $\underline{x}$ to the QP problem is guaranteed to be an extreme point, and therefore, boolean. However, there are n! extreme points and there are no known algorithms better than branch and bound methods to decide which of the n! extreme points is the right one. The problem is indeed a difficult one!

We now go on to discuss some evidence that the quadratic assignment problem is ill-posed, and that partial enumeration (also called implicit enumeration) is essentially the fastest algorithm for the problem as posed. Naturally such a method would be too slow for most purposes.

Stephen Cook [3] introduced the concept of polynomial time to measure the difficulty of problems. A class of problem is solvable in polynomial time if there exists an algorithm which solves any member of the class within $P(n)$ steps on a Turing machine, where $P(x)$ is a polynomial and n is some reasonable measure of the length of the problem. For example, in the quadratic assignment problem, n could be the number of entries in the matrices A and B; but then it may as well be the dimension of A and B. Stephen Cook has shown [4] that a number of ideal machines which resemble contemporary computers more than does a Turing machine can be used in place of a Turing machine in the definition without changing the classes of problems that are solvable in polynomial time.

We say that one problem P_1 is polynomial reducible to another problem P_2 if the solvability in polynomial time of P_2 would imply the solvability in polynomial time of P_1. We now define a few problems. Such expressions as $F = (p_1 \,\&\, \sim p_2) \vee (p_3 \,\&\, p_4)$ will denote propositions, and the p_i are variables over the set $\{T, F\}$. The binary operators & and V obey the usual truth table, as does the unitary operator ~. The expression F is a tautology if F takes the value T for all values of the p_i. This particular F is in disjunctive normal (D.N.F.) form because it is of the form $F = R_1 \vee R_2 \vee \ldots \vee R_k$, where $R_i = S_{i1} \,\&\, S_{i2} \,\&\, \ldots \,\&\, S_{im_i}$ and each S_{ij} is of the form p_r or $\sim p_r$. The problem of determining tautologyhood of propositions in D.N.F. is probably not solvable in polynomial time, but it has never been proved. It is suspected that the tree search method, which requires 2^n operations, where n is $\sum_{i=1}^{k} m_i$ is closer to the theoretical limit than any polynomial $p(n)$. The major reason for thinking that D.N.F.

tautologyhood cannot be decided in polynomial time is a theorem by Stephen Cook [3] which states that this would imply that every problem solvable on a non-deterministic Turing machine in polynomial time would be soluble on an ordinary Turing machine in polynomial time. A non-deterministic Turing machine (N.T.M.) is able to make copies of itself whenever it has a decision to make, each copy taking a different course. Thus there may be 2^n Turing machines after n steps.

Practically any finite problem can be solved in polynomial time on an N.T.M.; therefore D.N.F. tautologyhood must be about the most difficult to determine, since D.N.F. solubility in polynomial time would imply solubility of almost everything else in polynomial time. We shall prove theorems showing that D.N.F. tautologyhood is polynomial reducible to the quadratic assignment problem! To do this, we must first introduce the subgraph isomorphism problem.

A graph is a set of n elements called vertices together with a set of certain of the unordered pairs of these vertices, called edges. The complete graph of order n is the unique graph with n vertices and $\binom{n}{2}$ edges. Two graphs G and H are isomorphic if their vertices may be labeled in such a way that $u_1, \ldots, u_n$ denote the vertices of G, $v_1, \ldots, v_n$ denote the vertices of H, and $\{u_i, u_j\}$ is an edge of G if and only if $\{v_i, v_j\}$ is an edge of H. It is an unsolved problem whether graph isomorphism can be determined in polynomial time. There are n! different ways of numbering the vertices. If a certain conjecture of Corneil [5] is true, then graph isomorphism can be determined in polynomial time. It has never been shown that D.N.F. tautologyhood is polynomial reducible to the graph isomorphism problem. For our purposes

therefore, we must seek a harder problem.

Let G be a graph with vertices v_1, v_2, ..., v_n and edges e_1, ..., e_m. A graph H is a subgraph of G if H is obtained from G by possibly deleting some of G's vertices and deleting their incident edges and possibly deleting additional edges. The subgraph isomorphism problem is to determine, given graphs G and H, whether H is isomorphic to some subgraph of G.

Theorem 3 The problem of determining tautologyhood of propositions in disjunctive normal form (D.N.F.) is polynomial reducible to the complete subgraph isomorphism problem.

Proof Let the given proposition be $F = R_1 V R_2 V \ldots V R_k$ where $R_i = S_{i1} \& S_{i2} \& \ldots \& S_{im_i}$ and the S_{ij} are metavariables which take the letters p_1, p_2, ... or the symbols $\sim p_1$, $\sim p_2$, ... as their values; here ~ denotes "not".

Let the graph G have vertices the $m = \sum_{i=1}^{m} m_i$ vertices v_{ij} for $1 \le j \le m_i$, $1 \le i \le k$; let v_{ij} be joined to v_{rs} if $i \ne r$ and if S_{ij} and S_{rs} are not opposing - that is (S_{ij}, S_{rs}) is not of the form $(p_t, \sim p_t)$ or $(\sim p_t, p_t)$ for any t. (Graph theorists will observe that G is a k-partite graph). It is now obvious that F is a tautology if and only if it is not the case that the complete k-graph K_k is isomorphic to a subgraph of G. To see this, suppose first that F is not a tautology. Then there is an assignment of truth values to p_1, p_2, ... which falsifies F. This assignment must falsify every R_i, and consequently for every i, some S_{ij_i} must be

---

false. Hence the subgraph of G having vertices v_{ij_i} is a complete k-graph.

We now suppose that G has a complete subgraph of order k. Since v_{ij} and v_{rs} are only joined when i is different from j, we may suppose that the vertices of the complete subgraph are v_{ij_i} $(1 \leq i \leq k)$. We then assign truth values to $p_1, p_2, \ldots$ so that S_{ij_i} $(1 \leq i \leq k)$ are falsified and we see that F is not a tautology.

Theorem 4 The complete subgraph problem is polynomial reducible to the quadratic assignment problem.

Proof Let G be a graph with n vertices and let k be a positive integer not exceeding n. We wish to determine whether G contains a complete subgraph of order k.

Let $A = \{a_{ij}\}$ be the adjacency matrix for G; that is, if the vertices of G are $v_1, v_2, \ldots, v_n$, then let a_{ij} be 1 if v_i and v_j are joined by an edge and 0 otherwise. Thus A is a symmetric n x n matrix with zeros on the diagonal. We then let B be the k x k adjacency matrix of a complete k-graph. That is,

$$b_{ij} = \begin{cases} 1 \text{ if } i \neq j \\ 0 \text{ if } i = j \end{cases}$$

and let $$\tilde{B} = \left(\begin{array}{c:c} B & 0 \\ \hdashline 0 & 0 \end{array}\right)$$ be n x n.

Let $M = \max_P \operatorname{tr}(AP\tilde{B}P^T)$ where the maximum is taken over all n x n permutation matrices P. Then G has a complete subgraph of order k if and only if $M = k(k-1)$, and the theorem is proved.

Theorems three and four show that the quadratic assignment problem is at least as hard as D.N.F. tautologyhood, and Cook's theorems tell us that that is very serious. One could discuss some ways of limiting the quadratic assignment problem to a special class of matrices so that D.N.F. tautologyhood is excluded. We shall merely show that limiting A and B to be so-called distance matrices does not exclude D.N.F. tautologies.

<u>Theorem 5</u> (Robert Ray III) Any quadratic assignment problem

$$\max_P \operatorname{tr} (APBP^T)$$

where A, B are non-negative and symmetric is reducible to another quadratic assignment problem (QAP)

$$\max_P \operatorname{tr} (CPDP^T)$$

where c_{ij} is the distance between x_i and x_j, d_{ij} is the distance between y_i and y_j, and $x_1, \ldots, x_n$, $y_1, \ldots, y_n$ are 2n points in (n-1)-dimensional euclidean space. We call this the geometric quadratic assignment problem (GQAP). Consequently D.N.F. tautologyhood is polynomial reducible to GQAP. This result is due to Robert Ray III [6] and is part of his work on QUASCO, a heuristic program to solve the quadratic assignment problem.

<u>Proof</u> Let $E = \{e_{ij}\}$, $e_{ij} = 1$, the matrix of all ones. The following identifies are useful:

Let α be any real number.

(i) $\operatorname{tr}\{(A+\alpha I)PBP^T\} = \operatorname{tr}(APBP^T) + \alpha \operatorname{tr}(B)$

(ii) $\operatorname{tr}\{AP(B+\alpha I)P^T\} = \operatorname{tr}(APBP^T) + \alpha \operatorname{tr}(A)$

(iii) $\operatorname{tr} \{(A+ E)PBP^T\} = \operatorname{tr}(APBP^T) + \alpha \operatorname{tr}(EB)$

(iv) $\operatorname{tr} \{AP(B+ E)P^T\} = \operatorname{tr}(APBP^T) + \alpha \operatorname{tr}(AE)$

Now let A and B be as in the statement of the theorem. It is well known that there exist constants α, β such that $C = A + \alpha E - \alpha I$ and $D = B + \beta E - \beta I$ have the properties claimed and the identities (i) - (iv) show that $\max \operatorname{tr} (CPDP^T)$ will occur at the same P as $\max \operatorname{tr} (APBP^T)$, proving our theorem.

<u>Remark</u> There is one more useful identity (Robert Ray, III [6])

(v) $\operatorname{tr}\{(\alpha E-A)PBP^T\} = \alpha \operatorname{tr} (EB) - \operatorname{tr} (APBP^T)$.

If we apply this with $\alpha = \max\limits_{i,j} a_{ij}$ and put $\bar{A} = \alpha E-A$, then we get

$$\max_P \operatorname{tr} (\bar{A}PBP^T) = \max_P\{\alpha \operatorname{tr}(EB) - \operatorname{tr} (APBP^T)\}$$

or

$$\max_P \operatorname{tr} (\bar{A}PBP^T) = \alpha \operatorname{tr}(EB) - \min_P \{\operatorname{tr} (APBP^T)\}.$$

Hence every maximizing QAP is equivalent to a minimizing QAP (and vice versa).

References

[1] Nonlinear and Dynamic Programming, G. Hadley, 1964, Addison-Wesley.

[2] "There cannot be any algorithm for integer programming with quadratic constraints," Mimeographed notes by Robert G. Jeroslow, University of Minnesota. Received December 1971.

[3] Stephen Cook, Proceedings of the Third Annual ACM Symposium on the Theory of Computing, Shaker Heights, Ohio, 1970, p. 151-158

[4] Stephen Cook and Robert A. Reckhow, "Diagonal Theorems for Random Access Machines," University of Toronto, Computer Science Department report number 42, June 1972.

[5] D. G. Corneil, "Graph Isomorphism," University of Toronto, Computer Science Department report number 18, April 1970.

[6] Robert Ray, III, oral communication.

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D		
(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)		
1. ORIGINATING ACTIVITY (Corporate author) Center for Advanced Computation University of Illinois at Urbana-Champaign Urbana, Illinois 61801	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	2b. GROUP
3. REPORT TITLE The Quadratic Assignment Problem		
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Research Report		
5. AUTHOR(S) (First name, middle initial, last name) George B. Purdy		
6. REPORT DATE February 1973	7a. TOTAL NO. OF PAGES 19	7b. NO. OF REFS 6
8a. CONTRACT OR GRANT NO. DAHC04 72-C-0001	9a. ORIGINATOR'S REPORT NUMBER(S) CAC Document No. 71	
b. PROJECT NO. ARPA Order No. 1899		
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
d.		
10. DISTRIBUTION STATEMENT Copies may be requested from the address given in (1) above.		
11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY U. S. Army Research Office - Durham Duke Station, Durham, North Carolina	

13. ABSTRACT

In this report we discuss some seemingly reasonable approaches to the quadratic assignment problem, and we give some evidence from automata theory that the problem is insoluble.

DD FORM 1 NOV 65 1473

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Metatheory						
Graph Theory						
Non-Linear Programming						